IOU-Aware Multi-Expert Cascade Network via Dynamic Ensemble for Long-tailed Object Detection

Wan-Cyuan Fan*
National Taiwan University
r09942092@ntu.edu.tw

Cheng-Yao Hong*
Academia Sinica
sensible@iis.sinica.edu.tw

Yen-Chi Hsu
Academia Sinica
yenchi@iis.sinica.edu.tw

Tyng-Luh Liu
Academia Sinica
liutyng@iis.sinica.edu.tw

Abstract

*Object detection over a long-tailed large-scale dataset is practical and challenging, which is, however, under-explored comprehensively. Recently proposed methods mainly focus on eliminating the imbalanced classification problem. However, they do not take the quality of the predicted bounding boxes into consideration. Inspired by observation in Cascade architecture, "**detectors with different IOU thresholds have each favor to different quality of bounding boxes**", this paper first reveals the issue due to the unbalanced data distribution: the predicted bounding boxes' accuracy will be essentially different between categories. Thus, for example, the detector may predict inaccurate bounding boxes on the categories with fewer training data, and the corresponding extracted visual features will further damage the classification accuracy. To overcome such a problem, we introduce a Multi-Expert Cascade framework (MEC), a novel IOU-aware detector that re-weights each category's training process on different stages and achieves a better stage ensemble performance by leveraging dynamic ensemble mechanisms at the inference time. Extensive experiments on the recent long-tailed large vocabulary object detection dataset show that our proposed MEC framework significantly improves the most widely-used detectors' performance over various backbones on object detection and instance segmentation tasks.*

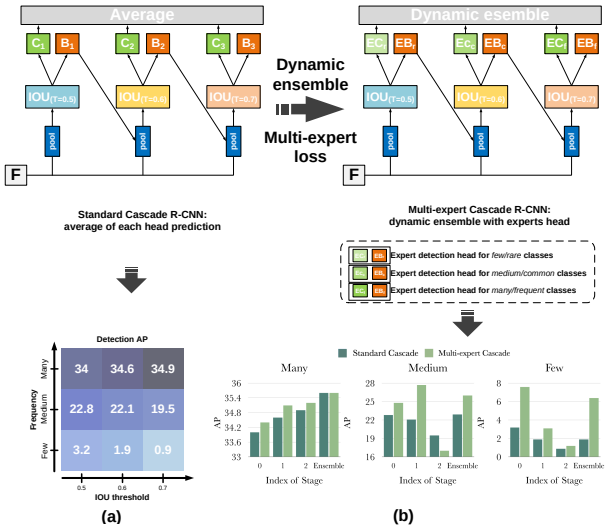

Figure 1. (a) shows that the performance of Cascade R-CNN of different frequencies on different stages. Besides, it points out that there is an imbalanced detection problem in the multi-stage detector. (b) presents the performance comparison between Cascade R-CNN and Cascade R-CNN with our approach (Both with ResNet101 as the backbone).

1. Introduction

Object detection is one of the most significant and challenging branches of computer vision tasks, which has wide applications in our daily life, e.g., robotics, monitoring security, autonomous driving. The goal of the object detection task is to recognize and locate objects of a set of pre-defined categories. Many of the recently proposed object detectors [43, 48] achieve promising results on some well-known benchmarks such as COCO [29] and PASCAL VOC [12]. These datasets are created by carefully selected, and the number of training samples of each category is relatively balanced. However, the data tends to be highly imbalanced in the real world, with a very long-tailed class distribution. Under this circumstance, many existing architectures may fail to achieve expected performance.

As for the object detection task, the challenge of training detectors on a long-tailed dataset largely derives from two

aspects. The first one is the imbalanced classification problem. This problem comes from the biased distribution of data across the known categories, which causes the classifier to remain under-trained on tailed categories with fewer data and makes the model tend to bias towards head categories (classes with numerous training samples). The second one is the imbalanced detection problem. Due to the insufficient training on the categories with fewer training data, the detector tends to predict the bounding boxes with unideal accuracy. Such inaccurate bounding boxes will further damage the classification performance.

In recent years, many approaches have been proposed to solve the imbalanced classification problem, such as [16, 21, 22, 24, 42, 47]. First, Agrim *et al*. [16] introduces a dynamic sampling factor that increases the probability to sample images with rare data during training. After that, Kang *et al*. [23], Zhou *et al*. [47], and Li *et al*. [24] point out that the image-wise re-sampling methods will damage the representation, and then they decouple the learning procedure into representation learning and classification learning. Recently, Tang *et al*. [42] argued that the imbalanced classification problem comes from lousy momentum causal effect and proposed de-confounded training and total direct effect inference to eliminate the causal effect.

Although many works proposed methods to solve the imbalanced classification problem, the imbalanced detection problem still has not been revealed. We take the architectures of Cascade mechanism [5, 8] which are perennial winners on the leaderboard as the starting point and provide some interesting observations. As shown in Figure 1 (a), we train various detectors with different foreground thresholds on a long-tailed dataset, and the result shows that the performance of each category will depend on the IOU threshold. For example, rare (categories with fewer training samples) instances perform better when the detector is trained with a lower IOU threshold, while frequent instances are the opposite. We argue that, at testing time, it is hard for a detector to predict the bounding boxes with high accuracy for rare instances. This induces that the detector trained with a lower IOU threshold achieves better performance for rare categories. In conclusion, there is an IOU-aware problem when the Cascade architectures [5, 8] face the instances' imbalanced distribution.

This paper proposes a novel multi-expert and IOU-aware detection framework called Multi-Expert Cascade (MEC) to address the imbalanced detection problem. The first key component of the MEC framework is multi-expert. First, we divide the classifier into multi-group according to the categories' amount of data. After that, we train the multi-stage detector with the expert loss, which re-weights the gradient in the training process for each category on different stages so that each classifier can emphasize specific categories. The second component, a dynamic ensemble mechanism to control the ensemble weights between these expert classifiers in the inference time to improve the effectiveness of the "expert" detectors. A simple comparison between standard Cascade R-CNN and our Multi-Expert Cascade on LVIS is provided in Figure 1 (b). Finally, we characterize the main contributions of our method to long-tailed object detection as follows:

- We unveil the imbalanced detection problem which means different frequencies of each category are sensitive at different heads in a multi-stage detector architecture.
- We propose a novel end-to-end framework, Multi-Expert Cascade (MEC) to tackle the imbalanced detection problem. MEC consists of two components: the multi-expert loss for training and the dynamic ensemble mechanism at inference.
- To produce an IOU-aware multi-stage detector, we utilize a unique multi-expert loss in the training process and learn the classifiers with different IOU thresholds equipping with the ability to be an expert on specific categories.
- The proposed dynamic ensemble mechanisms can exert the advantage of MEC by dynamically controlling the weight for each group in the classifiers in the inference phase to achieve better performance.
- The proposed method for lond-tailed detection achieves state of-the-art experimental results over existing techniques on two standard benchmark datasets, LVIS [16] and COCO-LT [24].

2. Related work

Object detection With the evolution of convolution neural networks, a great deal of success has been achieved in object detection [19, 39]. According to the characteristics, it can be divided into two categories. Unlike the one-stage based detector, which features real-time and efficient [30, 33–35], the most state-of-the-art detector in the detection task follows a two-stage regime, proposed region proposal, and classification. R-CNN series [14, 15, 18, 36] provided promising results on object detection. Based on that, some popular frameworks [5, 8] further improve the performance by multi-detection-head with proposed region refinement, iteratively.

Distribution re-balancing and losses re-weighting Re-sampling and Re-weighting are two common methods to alleviate the impact of the unbalanced data. Re-sampling in the early studies includes under-sampling [11] for head categories, and over-sampling [7, 17, 31] for tail categories. Recently, Shen *et al*. [37] proposed class-balanced sampling to weight the sampling frequency of each image accord-

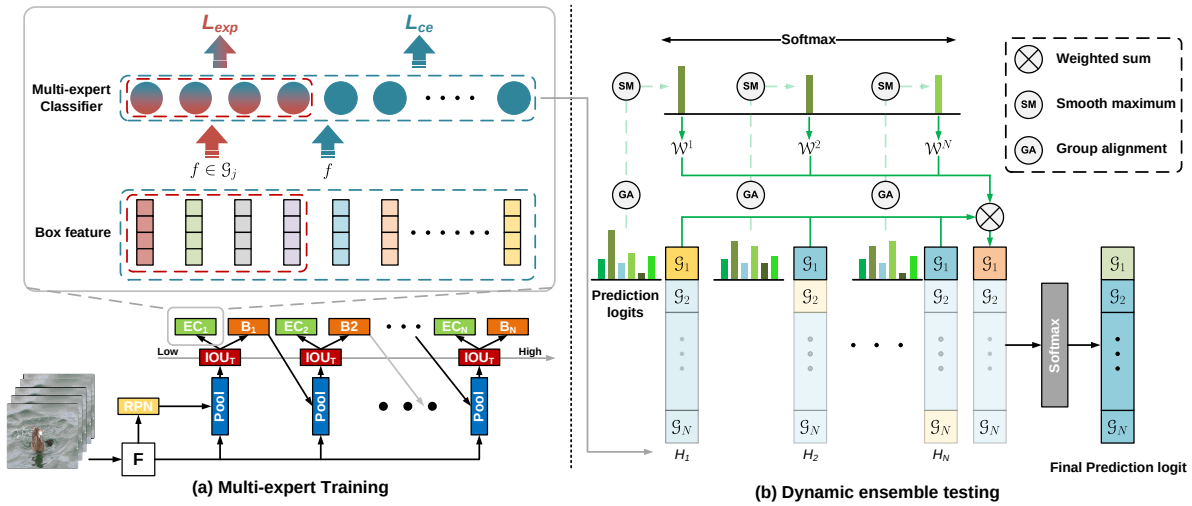

Figure 2. MEC consists of two components. (a) In the training phase, we leverage the observation that different orders of magnitude of categories favor different IOU thresholds to optimize each detector as the *multi-expert loss*. (b) In the inference process, we utilize an *dynamic ensemble mechanism* to exert the advantages from MEC training effectively.

ing to the number of samples of different categories, and Gupta *et al*. [16] proposed repeat factor sampling (RFS), a dynamic-sampling mechanism, to balance the instances. However, re-sampling is not a reliable solution. The tail categories are often learned repeatedly, lacking sufficient sample differences and not robust enough, and the head categories are often not sufficiently learned. Re-weighting methods, such as Hard Example Mining [38], Focal loss [27], and LDAM [6], is mainly adopted in the loss of classification by re-weighting based on category distribution. Unlike sampling, because of the flexibility and convenience of loss calculation, many more complex tasks, such as object detection and instance segmentation, are more likely to leverage the re-weighted loss to solve the unbalanced problem. Furthermore, due to implementation is easy, some works [10, 21, 40] show competitive results in complex tasks.

Training strategies for long-tailed representation learning Recent works [23,25,47] showed that the re-balancing methods damage the representation learning since changing the data distribution. Kang *et al*. [23] divides the learning process into two steps. The first step is using raw data (unbalanced) for representation learning. Furthermore, the second step leverages the class-balanced sampling mechanism for classifier learning. Based on the same assumption, Zhou *et al*. [47] transforms the two-step learning into a two-branch model to achieve an end-to-end training schema. Finally, one study adds another classifier to calibrate prediction logits [46]. Unlike the two-stage training strategies, Hu *et al*. [20] regards the learning of long-tail distribution data as a form of incremental learning, which learns common data first and then learns to recognize the rare categories based on previous knowledge. Also, Tang *et al*. [42] counts the moving average vector of a feature based on the traditional training framework, and this average feature will not participate in the gradient calculation during the end-to-end training. Recently, Li *et al*. [25] proposed the BAGS, which calculate only for the categories with the same order of magnitude instead of for all categories, to achieve a better-balanced classification learning. BAGS mainly focuses on designing classifiers to account for data imbalance when solving long-tail tasks. It does not explicitly address the problem of data imbalance of the multi-stage detector. However, we instead observe the detection imbalance problem—different orders of magnitude of categories favor different IOU thresholds, as shown in Figure 1. We are thus motivated to design the MEC to account for such detection imbalance problems for multi-stage detectors.

3. Methodology

In this section, we introduce our Multi-Expert Cascade (MEC) for addressing the imbalanced detection problem. Firstly, we begin by revisiting the standard multi-stage detector's preliminary in Section 3.1, the well-known Cascade R-CNN. Next, going through the MEC approach, which consists of two critical components: multi-expert loss and dynamic ensemble mechanism in Section 3.2 and Section 3.3, respectively.

3.1. Preliminary

Cascade R-CNN [5] is the well-known fundamental multi-stage detection model. In Cascade R-CNN, they de-

compose the difficult regression task into a sequence of simpler steps, and each step focuses on a different IOU proposal. In this way, Cascade R-CNN prevents the detection head from heavily tilting toward low-quality classifiers. In the stage *t*, the R-CNN includes a classifier h^t and a box regressor optimized for IoU threshold u^t, where $u^t > u^{t-1}$. Therefore, given a training set (x_i, y_i), where x_i is the roi feature of i-th bounding box and y_i is the corresponding ground truth class label, we can train the model by minimizing a classification cross-entropy loss as $L_{cls}(h^t(x_i), y_i) = CrossEntropy(h^t(x_i), y_i)$.

In the inference time, due to each stage focuses on a different IoU range, we empirically ensemble these stages to get higher performance. To be more specific, once the prediction logits o_i^t of i-th roi feature in the head t from classifier $h^t(x_i)$ obtain, we can formulate the ensemble score S_i in the inference phase as $S_i = \frac{1}{T}\sum_{t=1}^{T}\sigma(o_i^t)$, where σ is the Softmax activation function.

As shown above, we notice that, in Cascade R-CNN, the classification loss for each stage is identical, and the ensemble weight for each stage is equal. However, as the discussion in Section 1, we observe that categories with different amounts of training data may have different IOU thresholds favors. For example, rare data may perform better in the stage with a lower IOU threshold and perform poorly in the stage with a higher IOU threshold since the predicted bounding box has low precision. Therefore, using identical classification loss on categories for each stage and equalizing the predicted scores in the inference time may impede the final prediction or even damage the performance.

3.2. Multi-expert cascade formulation

When the distribution of categories is relatively imbalanced, *e.g.* a long-tailed dataset, the performance of these categories is highly correlated with the IOU between proposal and ground truth bounding box, which is shown in Figure 1 (a). To utilize this characteristic, we propose the Multi-Expert Cascade network, which enhances the positive and negative gradient for particular categories to equip each stage with the ability to focus on specific categories. Our framework architecture is illustrated in Figure 2.

For a multi-stage model H with N stages, we define H_k to represent k-th stage. After that, we divide all the categories into N groups according to the amount of their training instance. We assign categories i in $\mathcal{G}_n$ (n-th group) if $l_n \leq \delta(i) < l_{n+1}, \quad n > 0$, where $\delta(i)$ is the number of instance in training set for category i, and l_n and l_{n+1} are hyper-parameters to determine minimal and maximal number of instance for group n, respectively. For clear explanation later, we further define j-th group set G_j as a set containing $\mathcal{G}_j$ of all the stages. Following the setting in BAGS [25], we set $l_1 = 0$, $l_2 = 10^1$, $l_2 = 10^2$ and $l_3 = +\infty$. Also, throughout this paper, we set $N = 3$. Under this setting, $\mathcal{G}_1$ contains the categories with the amount of training data from 0 to 10, and G_1 is 1-st group set which includes three $\mathcal{G}_1$ from H_1, H_2, and H_3.

Inspired by the observation in Figure 1 (a), we define our expert group $\mathcal{G}_i^e$ as the group $\mathcal{G}_i$ in the stage H_i. In each stage, we enforce the classifier to pay more attention to the expert group. To achieve this idea, we make an additional prediction from each stage. In simple terms, given an input instance $x \in \mathbb{R}^d$, we will have the outputs from each stage as $o^{H_k} \in \mathbb{R}^C, k = 1, ..., N$. Furthermore, we can define the conventional prediction from the k-th stage $p_i^{H_k}$ and the prediction within the group $\tilde{p}_i^{H_k}$ as follows:

$$p_i^{H_k} = \frac{e^{o_i^{H_k}}}{\sum_{j=1}^{C} e^{o_j^{H_k}}} \text{ and } \tilde{p}_i^{H_k} = \begin{cases} \frac{e^{o_i^{H_k}}}{\sum_{j\in\mathcal{G}_k} e^{o_j^{H_k}}} & , i \in \mathcal{G}_k \\ 1 & , i \notin \mathcal{G}_k \end{cases}. \tag{1}$$

After that, our expert prediction can be formulated as follows:

$$\hat{p}_i^{H_k} = \begin{cases} (p_i^{H_k})^{\lambda}(\tilde{p}_i^{H_k})^{1-\lambda} & , i \in \mathcal{G}_k \\ p_i^{H_k} & , i \notin \mathcal{G}_k \end{cases} \tag{2}$$

with hyper-parameter λ which used to control the weight between $p_i^{H_k}$ and $\tilde{p}_i^{H_k}$.

To maximize the probability of expert prediction $\hat{p}_i^{H_k}$, we apply the cross-entropy loss on it and the multi-expert loss becomes $\mathcal{L}_{\text{exp}} = -\sum_{k=1}^{N}\sum_{i=1}^{C} y_i \log \hat{p}_i^{H_k}$.

Through the equation (2), we can successfully strengthen the performance of each head on each expert group by adding probability to re-weight the original cross-entropy loss. As shown in Figure 2, take H_1 for an example, we combine an external cross-entropy loss on H_1 within the rare group $\mathcal{G}_1$ when the input sample belongs to the rare categories. This multi-expert loss will force the H_1 to not only make a correct prediction against the whole categories but also predict accurately at the rare group $\mathcal{G}_1$.

3.3. Group alignment and dynamic ensemble

We first apply group alignment (GA) to the classifiers' predicted logits in the inference phase and then utilize a dynamic ensemble mechanism to get the final prediction scores. We will go through the details of these modules as follows.

Group alignment In MEC, the multi-expert loss equips our classifier with the ability to better distinguish the categories in each expert group by enlarging the difference between the maximum value and the average value. Therefore, we perform group alignment to calculate this difference which will be set as the new predicted logits for the further ensemble mechanism if category i in the group $\mathcal{G}_j$,

the logit of this categories $\hat{o}_i$ after the group alignment can be formulated as follows.

$$\hat{o}_i = o_i - \frac{1}{N_{\mathcal{G}_j}} \sum_{i \in \mathcal{G}_j} o_i, \tag{3}$$

where $N_{\mathcal{G}_j}$ is the number of categories in group $\mathcal{G}_j$.

Dynamic ensemble inference As illustrated in Figure 2 (b), after applying group alignment, we try to ensemble these stages to achieve better performance. Instead of directly ensemble by an average self-ensemble as standard multi-stage such as architecture [5, 8], we propose two ensemble approaches. One is the sparse ensemble mechanism, and the other is the dynamic ensemble mechanism. Different from the average self-ensemble, both of them can effectively exert the advantages from Multi-Expert cascade training. For the sake of introducing them simply, we formulate the logit of class i in the head H_k as $o_i^{H_k}$ and define the logit set in a group as $O_{\mathcal{G}_j}^{H_k} = \{o_i^{H_k} | i \in \mathcal{G}_j\}$, where group $\mathcal{G}_j$ is the class set of j-th group, and $k = 1, ..., N$.

First, the sparse ensemble mechanism is an intuitive mechanism. It believes that each head is an expert at the group we are assigned. Hence, the output is composed of three different experts sparsely. The inference output becomes

$$o_i^{\text{sparse}} = \sum_{k=1}^{N} e_i^k o_i^{H_k}, \text{ where } e_i^k = \begin{cases} 1 & , i \in \mathcal{G}_k \\ 0 & , i \notin \mathcal{G}_k \end{cases}. \tag{4}$$

Secondly, the dynamic ensemble mechanism tries to utilize the characteristics of our Multi-Expert Cascade model smoothly. In other words, the multi-stage module now can better distinguish the category of the input instance in each expert group by enlarging the difference between the highest logit and the mean of all logits. As shown in Figure 2 (a), to obtain the highest logit in each group, we first exploit the LogSumExp (LSE) as the smooth maximum function, and then the logit of i-th category on k-th stage will become

$$o_i^{\text{dynamic}} = \sum_{k=1}^{N} w_i^k o_i^{H_k}, \text{ where } w_i^k = \frac{e^{LSE(O_{\mathcal{G}_j}^{H_k})}}{\sum_{k=1}^{N} e^{LSE(O_{\mathcal{G}_j}^{H_k})}}, \ i \in \mathcal{G}_j \tag{5}$$

and

$$LSE(O_{\mathcal{G}_j}^{H_k}) = \log \sum_{i \in \mathcal{G}_j} e^{o_i^{H_k}/\tau} \tag{6}$$

with temperature τ. Different from the sparse ensemble mechanism, the dynamic ensemble mechanism will consider the weight from each head; Moreover, it is between the average self-ensemble and the sparse ensemble mechanism.

After the ensemble mechanism, the output probability vector can now be formulated as $S^{\mathcal{E}} = \sigma(o^{\mathcal{E}})$, where σ is a softmax activation function and $\mathcal{E}$ means the ensemble mechanisms we proposed. Furthermore, it will be fed to the following post-processing steps such as NMS to make the final detection results.

4. Experimental Settings

Datasets. We perform extensive experiments on the recent Large Vocabulary Instance Segmentation (LVIS) dataset [16] and **COCO-LT** [25] dataset. (i) **LVIS v1.0** contains 100,170 training images and 19,809 validation images. Furthermore, LVIS contains 1,203 categories with both bounding box and instance mask annotations. These categories are divided into three groups based on the number of images that contain those categories: rare (1-10), common (11-100), and frequent (>100 images). In the evaluation phase, We use official metrics mAP as the measurement metric. Also, we calculate AP for each group: AP_r (AP for rare classes), AP_c (AP for common classes), and AP_f (AP for frequent classes). (ii) To validate methods on the dataset with long-tailed distribution, we create **COCO-LT** dataset, which is a subset of COCO, by following the construction methods in BAGS [25]. COCO-LT follows long-tail distribution just like LVIS and contains 16,966 training images of 80 categories, including 128,615 training instances. More details about the datasets are provided in the supplementary materials.

5. Results and Analysis

5.1. Effectiveness of MEC

This section demonstrates the effectiveness of our proposed framework by applying MEC in two different settings: (*i*) different backbones and frameworks. (*ii*) different classification methods on the long-tailed dataset. There is a detailed analysis of the results with reference to Table 1.

Is the proposed method flexible in other frameworks? Yes. The MEC can adapt effectively to different backbones and frameworks. (*i*) We apply our MEC to the well-known multi-stage model such as Cascade R-CNN and hybrid task cascade (HTC). As shown in Table 1, our model consistently improves the performance on all models, especially on rare and common categories. Take model (3) and (4) for example. With the help of our MEC module, the baseline Cascade R-101 model achieves +4.3 improvement on the rare mask AP. (*ii*) We incorporate our MEC with recently proposed long-tailed classification methods, most of which focus on eliminating the classifier imbalanced problem. As shown from model (9) to (14) in Table 1, our method stills improve each model significantly (rare categories in particular), which verifies that our model does not conflict with

ID	Model	Backbone	MEC	AP	AP_r	AP_c	AP_f	mAP^m	AP^m_r	AP^m_c	AP^m_f
(1)	Cascade R-CNN† [5]	R-50-FPN	×	22.1	1.4	19.6	**34.3**	19.7	1.0	18.0	**29.9**
(2)			✓	**25.0**	**5.5**	**24.5**	34.1	**22.3**	**5.4**	**22.3**	29.8
(3)	Cascade R-CNN† [5]	R-101-FPN	×	24.2	1.9	22.9	**35.6**	21.6	1.8	20.8	31.1
(4)			✓	**26.4**	**6.4**	**26.0**	35.5	**23.5**	**6.1**	**23.6**	**31.1**
(5)	HTC† [8]	R-50-FPN	×	23.3	1.4	20.3	**35.2**	20.4	1.4	19.3	31.3
(6)			✓	**25.6**	**6.1**	**23.9**	35.1	**22.8**	**5.7**	**22.8**	**31.4**
(7)	HTC† [8]	R-101-FPN	×	24.2	1.9	21.9	**36.7**	22.1	1.5	20.7	32.7
(8)			✓	**27.1**	**6.8**	**26.5**	36.6	**24.9**	**6.6**	**25.1**	**32.7**
(9)	SeesawLoss‡ [44]	R-50-FPN	×	24.3	2.9	23.1	33.5	21.4	2.5	21.7	29.6
(10)			✓	**25.4**	**6.4**	**24.5**	**34.2**	**22.3**	**5.1**	**22.7**	**30.2**
(11)	Cos-Norm‡ [13]	R-50-FPN	×	25.6	7.1	24.5	**34.8**	22.6	6.5	22.0	**30.3**
(12)			✓	**28.9**	**18.5**	**28.2**	34.1	**25.9**	**17.2**	**25.9**	29.6
(13)	De-confound† [42]	R-50-FPN	×	28.2	15.7	27.4	**34.6**	25.0	14.2	24.6	30.1
(14)			✓	**28.6**	**17.4**	**27.8**	34.5	**25.3**	**15.8**	**24.8**	**30.0**

Table 1. Comparasion of the performance gain brought by the MEC and other long-tailed classification methods on multiple backbone networks on LVIS v1.0. Note that AP^m denotes the score on the instance segmentation task. The † denotes reproducing the results from their official code, and ‡ represents re-implementing by us. Also, ✓ and × mean the method is combining with MEC or not.

other SOTA methods. These results signify the effectiveness of our Multi-Expert cascade network on a long-tailed distribution dataset and indicate the compatibility of our MEC with other long-tailed classification methods.

5.2. Comparison with other LT methods

In this section, we conduct experiments on the LVIS v1.0 dataset [16] and comparing with multiple state-of-the-art methods, which mainly tackle the classification imbalance problems, including repeat factor sampling (RFS) [16], Focal Loss [27], Equalization Loss [40], Seesaw Loss [44], Classification Calibration Head [45], De-confound [42], and BAGS [25]. For the fair comparison, we report all the results on object detection and instance segmentation with Cascade R-CNN with R-50-FPN as the backbone in Table 2.

Can we learn better representation by MEC? As illustrated in Table 2 (a), we observe that our method not only outperforms the conventional re-sampling/re-weighting (model (2-5)) but also exceeds the recent proposed gradient re-weighting methods such as EQ Loss (model (6)) and seesaw loss (model (7-8)) by a significant margin, especially on rare and common categories. The current methods, BAGS and De-confound, are shown in the model (10, 11). For model (10), even though the performance on rare and frequent data of our MEC is slightly lower than BAGS, our methods surpass the BAGS by 1.6 on common categories. The final mask mAP score accordingly outperforms the BAGS by 0.5. For (11), the De-confound-TDE method improves AP on rare, common, and frequent categories. Nevertheless, De-confound still has room for improvement on rare categories because of not considering the impact of the imbalanced detection problem. These comparison results not only verify that a better representation can be obtained by our MEC but also points out tackling imbalanced detection problem is necessary for object detection task on such imbalanced dataset. Furthermore, as the results are shown in both Figure 3, our MEC predicts a more detailed instance than the standard cascade model. For example, our MEC successfully segments the"life jacket", "wet suit", "raft" and "kayak" (*common category*), in the contrast, the standard cascade model is misclassified as background. Again, these examples verify that our proposed model can predict *common/rare* objects better than the basic cascade model.

How well does our method perform? Our MEC results are shown in model (12), which boosts the AP_r and AP_c significantly and slightly improves the AP_f by 0.4. Since our method tackles the imbalanced detection problem and does not focus on the classifier weight imbalance. We report our MEC with cosine classifier as same as the seesaw loss in the model (8). The proposed method leverages the discovery as mentioned in Figures 1, and the overall results outperform other long-tailed methods signify the essence of tackling the imbalanced detection problem.

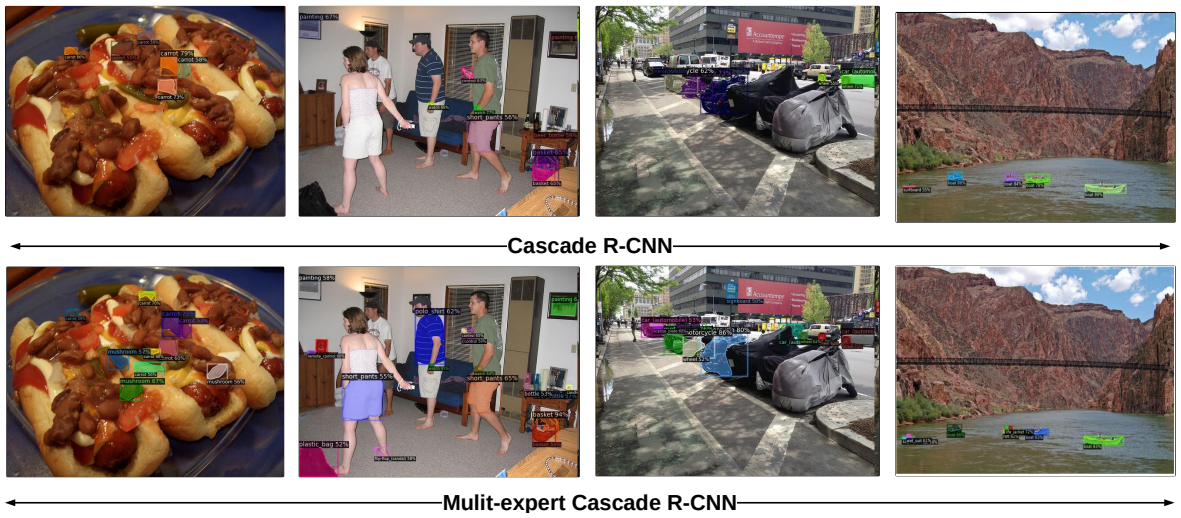

Figure 3. **Qualitative comparison on LVIS v1.0.** The visual results demonstrates the proposed method (*Bottom*) and basic Cascade R-CNN (*Top*) on a dense instance segmentation example containing *common/rare* category.

ID	Models	**mAP**^{m}	AP^m_r	AP^m_c	AP^m_f	**mAP**
(1)	Baseline	19.7	1.0	18.0	29.9	22.1
(2)	RFS† [16]	23.0	10.4	24.4	29.5	26.5
(3)	RFS-cls† [16]	23.0	9.5	24.1	29.4	25.7
(4)	Focal loss‡ [27]	10.4	0.8	5.1	20.6	11.2
(5)	Focal loss-cls‡ [27]	17.2	0.9	14.2	27.88	18.7
(6)	EQ loss† [40]	21.6	4.1	22.1	28.4	24.6
(7)	seesaw loss‡ [44]	21.4	2.5	21.7	29.6	24.3
(8)	seesaw loss-cos‡ [44]	24.8	13.6	25.9	**30.5**	27.7
(9)	CChead† [45]	21.5	13.2	18.3	29.6	24.5
(10)	BAGS† [25]	25.6	17.2	25.7	29.3	28.8
(11)	De-confound-TDE† [42]	25.0	14.2	24.6	30.0	28.2
(12)	MEC	**26.1**	**17.4**	**26.7**	29.6	**29.0**

(a)

Model	**mAP**^{m}	AP$_{50}$	AP$_{75}$	AP^m_r	AP^m_c	AP^m_f	**mAP**
Baseline	14.0	24.5	14.3	3.8	17.7	20.0	16.7
EQLoss† [40]	12.8	22.1	13.0	3.3	14.2	**20.3**	15.2
SeesawLoss‡ [44]	14.1	24.6	14.3	3.8	**18.2**	19.8	16.8
MEC	**14.5**	**24.9**	**14.5**	**5.0**	18.0	19.8	**17.1**

(b)

Model	Exp.	DI	SI	GA	**mAP**^{m}	AP^m_r	AP^m_c	AP^m_f	**mAP**
R50					19.7	1.0	18.0	29.9	22.1
		✓		✓	21.0	3.7	19.6	**30.1**	23.5
	✓				20.5	1.4	19.6	29.9	22.9
	✓	✓			21.5	3.6	21.1	29.8	24.0
	✓			✓	21.3	2.9	20.6	**30.2**	23.9
	✓		✓	✓	21.7	5.2	21.9	28.7	24.2
	✓	✓		✓	**22.3**	**5.4**	**22.3**	29.8	**25.0**

(c)

Table 2. (a) Comparison with STOA methods on LVIS val set. (b) The results on COCO-LT. (c) Ablation studies of MEC on LVIS val set. The Exp., DI, SI, and GA represent multi-expert loss, dynamic inference, sparse inference, and group alignment. We denote AP^m as average mask precision, for instance, segmentation. Also, note that † denotes reproducing the results from their official code, and ‡ represents re-implementing by us.

5.3. Evaluation on other LT object detection task

To further verify the generalization of our method, we build a COCO-LT dataset, which has similar long-tailed distribution as LVIS, by sampling images and annotation from COCO [29] dataset. The statistical results will be provided in our supplementary. As the results are shown in Table 2 (b), we observe that our MEC still improves the baseline method, especially on rare categories, which confirms our MEC's ability on the long-tailed dataset.

5.4. Ablation Studies

To elaborate MEC, we perform several ablation studies in this section. For the following experiments, our default model is the Cascade R-CNN with R50-FPN backbone.

What is the gain from the each component of the MEC? Table 2 (c) lists the performances and compares contributions of the deployed modules in our Multi-Expert cascade network. To confirm our introduction and enforcement of

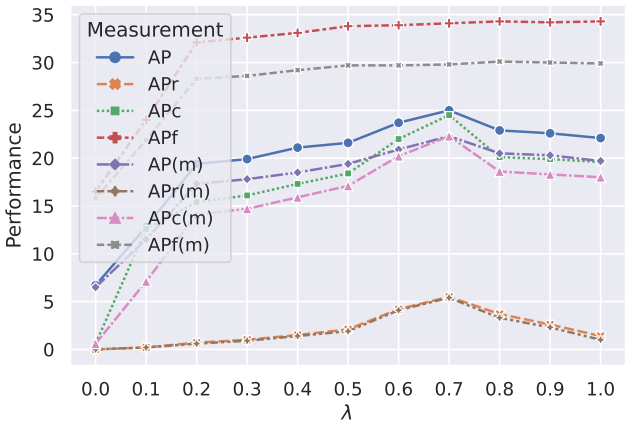

Figure 4. Ablation studies of different λ in the multi-expert loss on the LVIS validation set. Note that we use "(m)" to represent the mask average precision for instance segmentation.

Model	# of stage	mAP	AP_r	AP_c	AP_f	mAP^m	AP_r^m	AP_c^m	AP_f^m
Baseline	3	22.1	1.4	19.6	34.3	19.7	1	18.0	29.9
MEC	3	25.0	5.5	24.5	34.1	22.3	5.4	22.3	29.8
Baseline	4	22.3	1.1	19.6	34.6	19.8	1	18.1	29.9
MEC	4	26.1	6.9	25.8	34.9	23.2	6.7	23.3	30.2

Table 3. Experiments with various number of stages.

expert loss during training, we apply this objective function to the baseline model (first row) and report the results in the third row of Table 2 (c). Moreover, with the dynamic ensemble, the inference phase of our Multi-Expert Cascade network is enhanced, with results listed in the fourth row of Table 2 (c). Furthermore, we report the performance of the dynamic ensemble with group alignment in the fifth row of Table 2 (c). In the last two rows in Table 2 (c), we also demonstrate the performance comparison between two types of the proposed inference mechanism. Also, to verify the generalization of the dynamic inference and group alignment, we apply them on the cascade R-CNN baseline, as shown in the second row of Table 2 (c). The result shows that the methods improve the baseline model by 1.3 in mask mAP. Also, the export loss further improves such a baseline model from 21.0 to 22.3 in mask mAP. By comparing the performances listed in Table 2 (c), we see that the full version of our MEC achieved the best performance in terms of both object detection and instance segmentation. Thus, the design of our MEC can be successfully verified.

How is the intensity of expert loss "λ" affected? The influence of different expert threshold λ is shown in Figure 4. In this section, we utilize the dynamic ensemble for the testing. After that, we train our MEC with different λ on the LIVS dataset and report each model's performance. As a result shown in Figure 4, our Multi-Expert cascade achieves the best mAP score when λ = 0.7. Therefore, we use λ = 0.7 as our default setting for all the experiments.

6. Discussion

In this section, we provide some discussion to the common concerns. The experimental results echo our motivation and validate the proposed approach, MEC.

Dose our MEC improve the performance on rare categories mainly because the lower threshold stage allow more data sampled during training? To clarify this concern, we analyze a Cascade R-CNN baseline model and calculate the number of predicted bounding boxes of rare categories on each stage. We found that the average ratio of rare predictions for three stages is 1:1.008:1.078. In other words, after the box regression at each stage, the number of rare predicted boxes increases in general during the training process. Therefore, we set the rare expert head to the stage with a lower IOU threshold has nothing to do with the number of samples because each head has almost the same number of sampled data.

How is the generalized ability of MEC when applying on the detectors with different numbers of stages? In Table 3, we conduct the experiment on a 4 stages detector and compare it with the results of 3 stages detector. Note that we divide all the categories into 4 groups according to the method described in Section 3.2, and we set the $l_3 = 500$ and $l_4 = +\infty$. For both MEC models with 3 and 4 stages, we apply dynamic inference and group alignment in the testing phase. The results show that MEC achieves 3 and 4 improvements on mAP when using 3 stages and 4 stages, respectively. In this paper, we set our MEC baseline as 3 stages detector to lower down the computation cost.

7. Conclusions

This paper improves the multi-stage detector framework by the proposed multi-expert cascade (MEC) for long-tailed object detection. The proposed approach is specifically designed to address the issue of imbalanced detection and explore the subtle relationship of performance variation between IOU threshold and data frequency. The solid and consistent detection improvements of the MEC on the challenging LVIS and COCO-LT suggest that modeling the corresponding relationship between categories and their respective occurring frequency is required to advance long-tailed object detection. MEC is shown to be advantageous to several object detection architectures on long-tailed distribution datasets. We believe the introduction of MEC can help advance our understanding in better solving a broad range of long-tailed object detection tasks.

References

[1] Balancedgroupsoftmax-github. https://github.com/FishYuLi/BalancedGroupSoftmax. 2

[2] De-confounded-github. https://github.com/KaihuaTang/Long-Tailed-Recognition.pytorch. 2

[3] EQLoss-github. https://github.com/tztztztztz/eql.detectron2. 1

[4] SeesawLoss-github. https://github.com/bamps53/SeesawLoss. 1

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 2, 3, 5, 6

[6] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 1565–1576, 2019. 3

[7] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002. 2

[8] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4974–4983, 2019. 2, 5, 6

[9] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[10] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9268–9277, 2019. 3

[11] Chris Drummond, Robert C Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pages 1–8. Citeseer, 2003. 2

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1

[13] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 6

[14] Ross B. Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1440–1448, 2015. 2

[15] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587, 2014. 2

[16] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 2, 3, 5, 6, 7, 1

[17] Hui Han, Wenyuan Wang, and Binghuan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing, International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I*, pages 878–887, 2005. 2

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988, 2017. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. 2

[20] Xinting Hu, Yi Jiang, Kaihua Tang, Jingyuan Chen, Chunyan Miao, and Hanwang Zhang. Learning to segment the tail. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14042–14051, 2020. 3

[21] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 7607–7616, 2020. 2, 3

[22] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020. 2

[23] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020. 2, 3

[24] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10991–11000, 2020. 2

[25] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10988–10997, 2020. 3, 4, 5, 6, 7

[26] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. *CoRR*, abs/2006.10408, 2020. 2

[27] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007, 2017. 3, 6, 7

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 7

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, pages 21–37, 2016. 2

[31] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. 2

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019. 1

[33] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788, 2016. 2

[34] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6517–6525, 2017. 2

[35] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 2

[36] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015. 2

[37] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, pages 467–482, 2016. 2

[38] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 761–769, 2016. 3

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[40] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11659–11668, 2020. 3, 6, 7

[41] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11662–11671, 2020. 1

[42] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3, 6, 7

[43] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. *arXiv preprint arXiv:2011.08036*, 2020. 1

[44] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. *arXiv preprint arXiv:2008.10032*, 2020. 6, 7, 1

[45] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Jun Hao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. Classification calibration for long-tail instance segmentation. *arXiv preprint arXiv:1910.13081*, 2019. 6, 7, 1

[46] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Jun Hao Liew, Sheng Tang, Steven C. H. Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. *CoRR*, abs/2007.11978, 2020. 3

[47] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9719–9728, 2020. 2, 3

[48] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection, 2021. 1

Appendix

A. Implementation details

A.1. Model setup

For the experiments in the main paper, we implement cascade r-cnn R50 with FPN as our baseline model. In the training phase, input images are resized to multiple size with (1333, 640), (1333, 672), (1333, 704), (1333, 736), (1333, 768), (1333, 800). We do not apply other augmentation except the horizontal flipping. For the region proposal network (RPN), we use the default setting as sampling 256 anchors with a 1:1 ratio between the background and foreground. After that, 512 proposals are sampled per image with a 1:3 foreground-background ratio for the second stage. For the cascade head, we use three stages: box, class, and mask predictor for each stage. Also, we set the IOU threshold as 0.5, 0.6, 0.7 in the three stages, respectively. Our baseline model is optimized by stochastic gradient descent (SGD) with momentum 0.9 and weight decay 0.0001 for 20(LVIS)/12 (COCO-LT) epochs, with an initial learning rate of 0.02, which is decayed to 0.002 and 0.0002 at 16(LVIS) /8(COCO) epoch and 18(LIVS)/10(COCO) epoch respectively. All of the experiments are processed with 16 GPUs with a total batch size of 16. We resize the input images into (1333, 800) in the testing phase and only use *random filp* augmentation.

A.2. Experiment setup

The all implementations of our model are based on the MMDetection platform [9] and Pytorch [32]. All the models are trained with 8 Nvidia V100 GPUs, with a batch size of 2 per GPU. In all the experiments, the models are trained with 20 epochs, except for fine-tuned training methods that only trained the fully connected layers in classifiers such as (3) and (5) in Table 2. We use the SGD optimizer with a 0.02 learning rate and decays our learning rate at the 12th and 16th epochs with 0.1 factor. Also, we warm up our model at the first 1000 iteration by linearly increasing the learning rate from 0.002 to 0.02.

A.3. Implementation of other methods

Repeat factor sampling (RFS) [16] was proposed in the LVIS dataset original paper at the same time. In this section, we report the implementation details of other methods in Table 2.

Repeat factor sampling (RFS) We directly apply the RFS package in MMDetection platform. However, we apply two different training schemes to the RFS model: *(i)* End-to-end training with RFS (threshold = 0.001) *(ii)* Based on our baseline model (1), only fine-tuning the fully connected layer in classifiers of the prediction head with RFS (threshold = 0.001). The results of these models are shown in models (2) and (3) in Table 2.

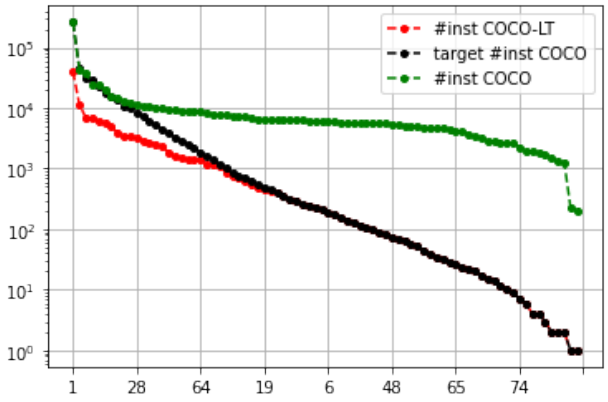

Figure 5. We align categories of COCO and LVIS and sample the corresponding number of instances for each COCO category. Note that, **red** line denotes the data distribution of COCO-LT dataset, **black** line indicates the corresponding number of instances for COCO dataset to matched the long-tailed distribution of LVIS, and **green** line represents the data distribution of the original COCO dataset. Also, note that their instance number sorts the categories.

Focal loss Focal loss [28] re-weights the cost at image-level for classification. Like the RFS method, we implement focal loss on the baseline model with two different training schemes: End-to-end and fine-tuning Fc in the classifier. In the end-to-end training, we directly apply sigmoid focal loss at the proposal level. In the fine-tuning method, we first initialize the model with the baseline model (1). Then we apply sigmoid focal loss on the classification head to fine-tune the only classifier (W, b).

EQ loss and seesaw loss EQLoss [41] and Seesaw loss [44] are similar methods for long-tailed classification, which reduce the negative gradient on tail classes by modifying on original cross-entropy loss. For EQLoss [41], we directly utilize the official code [3] and apply it to our baseline model to do end-to-end training. For seesaw loss, we use the unofficial implementation code in [4] to do end-to-end training.

Classification Calibration Head Classification calibration [45] is another training framework for long-tailed object detection. First, we initialize the model with our baseline model. After that, we fix the whole model's parameters except for CChead and train the classification calibration head for 12 epochs with the same setting as in [45]. Finally, in the inference phase, we combine the original classification head and the classification calibration head to achieve the final score for the prediction.

ID	Models	Backbone	**mAP**^{m}	AP^m_r	AP^m_c	AP^m_f	**mAP**
(1)	Baseline	Cascade R50-FPN	19.7	1.0	18.0	29.9	22.1
(2)	MEC-Net	Cascade R50-FPN	22.3	5.4	22.3	29.8	25.0
(3)	MEC-Net-cos	Cascade R50-FPN	25.9	17.2	25.9	29.6	28.9
(4)	MEC-Net-cos	HTC R101-FPN	27.9	18.6	29.7	31.4	31.8
(5)	MEC-Net-cos + SyncBN + DCN	HTC ResNeSt200-FPN	**35.4**	**23.4**	**34.1**	**40.8**	**38.1**

Table 4. The bounding box results and mask AP compared our method with other state-of-the-art methods on LVIS val set. Note that we denote AP^m as average mask precision, for instance, segmentation.

Balanced GroupSoftmax (BAGS) method BAGS [26] is a recently proposed method that tackles the imbalanced problem in the classifier via dividing the categories into many groups according to the number of instances and applying the softmax function independently for each group when fine-tuning the classifier. We directly use the officially released code from [1] to implement the model in Table 2. Firstly, We train the standard cascade r50 model for 20 epochs. Secondly, following the setting in BAGS [26], we erase the parameters of classifiers and then train the classifiers only while applying the GroupSoftmax for 12 epochs. The results are shown in model (10) in Table 2.

De-confounded TDE method De-confound [42] with TDE inference is a recently proposed method for long-tailed classification. This method contains two components: *(i)* De-confounded classifier *(ii)* Total Direct Effect Inference. We use the officially released code in [2] to reproduce the result on the LVIS dataset. We set the head as two for the de-confounded classifier, and the scale for input x is 8. Also, the weight of the causal norm is 1/32. After the end-to-end de-confounded training, we apply total direct effect inference with $\alpha = 1.5$. The results are shown in model (11) in Table 2.

B. Details about COCO-LT dataset

To confirm the effectiveness of our model, we construct the COCO-LT dataset to verify our MEC-Net further. To get a dataset with similar long-tail distribution as LVIS, we first sort all categories of LVIS and COCO by their corresponding number of training instances. As shown in Figure 5, we align the categories of the COCO and LVIS dataset and calculate the number of training instances per category in COCO-LT based on its corresponding categories in LVIS. After setting the target number of instances for each category, we apply image-level sampling on the COCO dataset to fit the target number for all categories as well as possible. However, because we apply image-level sampling, it is impossible to get an instance number for each category, which perfectly matches our target instance number. Therefore, we ignore the target instance number of the top 10 categories with the most instances. That is, the instance number of these classes might be less than their target instance number. In this way, other categories can perfectly match the target instance number by image-level sampling from the COCO dataset. The final distribution for the COCO-LT dataset is shown in Figure 5. COCO-LT only contains 16,966 training images of 80 categories, which includes 128,615 training instances. For validation, we use the same validation set as COCO val 2017 split, which includes 5000 images.

C. Additional Results

C.1. Results: Better backbone

In this section, to further confirm the performance of our MEC, we apply our method on LVIS with a better backbone and other training/testing tricks. The results are shown in Table 4.

Baseline We utilize the Cascade R-CNN with R50-FPN as the backbone for our baseline model (1). Other training settings are the same as the settings in Section A.1.

Multi-expert cascade network Our final result with better backbone and training tricks is shown in model (5) in Table 4. In this mode, we utilize HTC with ResNeSt200-FPN as our backbone and apply end-to-end training with DCN and SyncBN methods. We apply testing time augmentation (TTA) in the testing phase, including randomly flipping and multi-scale testing input. Finally, We achieved mAP 35.4 on the LIVS dataset. It is worth noting that the performance of tailed classes achieves 23.4 in mask AP score.